METHOD OF LOGIC CIRCUIT SYNTHESIS AND DESIGN USING A DYNAMIC CIRCUIT LIBRARY

TECHNICAL FIELD OF THE INVENTION

This invention relates to the design of digital logic circuits. More particularly, the invention relates to a method for synthesizing logic circuits from high-level logic representations.

BACKGROUND OF THE INVENTION

Logic circuits are electronic circuits that perform some logical operation or group of logical operations on a set of digital input data to provide some desired digital output. For example, a simple logic circuit may perform an AND operation on input data comprising three logic level signals to provide an single output comprising the logical signal resulting from the AND operation. These types of logic circuits are included in many different types of devices from control circuits for mechanical devices, to communications devices, to data processor chips. In many cases the logical operations may be very complicated and require complicated logic circuitry.

Logic circuit design involves first determining the particular logical function or operation which must be performed by the logic circuit. In this step, the designer works with high-level logic building blocks such as AND and OR gates to specify the logical

operations to be performed in the final circuit design. When determining the logical operations required in the ultimate circuit, the designer is generally not concerned with the actual circuit elements required to provide the high-level logical operation or functionality.

After determining the high-level logic operations required of the final logic circuit, logic circuit design involves determining an arrangement of circuit elements to actually perform the desired logical operations. The step of determining the actual circuit elements and arrangement of circuit elements to provide the desired logical operations is referred to as logic synthesis. Logic synthesis my be performed by defining a set or library of available circuits and then identifying an arrangement of these available circuits which is capable of consistently providing the desired logical operation. Thus, logic synthesis comprises an optimization problem constrained by the library of available circuits available for synthesis, by the required logical operation to be performed by the final circuit, and perhaps by other constraints dictated by the designer. Various software tools have been developed to perform logic synthesis. These logic synthesis tools simply provide one or more solutions to the optimization problem, based upon a library of available circuits and other constraints provided by the designer.

Prior logic synthesis tools employ circuit libraries containing many different types of circuits. These prior circuit libraries include not only numerous different circuits, but may also include numerous versions of the same circuit, employing different sized circuit elements. Utilizing a large circuit library provides a certain flexibility in developing the

ultimate circuit design. However, this flexibility comes at the cost of vastly increasing the complexity of the optimization problem which must be performed. This increase in complexity results in an increase in the processing time which the synthesis tool requires in order to produce a solution.

Another problem associated with prior static circuit logic synthesis involves the timing performance of the synthesized circuit. Timing between the signals applied to the input of the synthesized static circuit and the signals appearing at the output of the circuit is entirely dependent upon the manner in which signals propagate through the various static components of the circuit. Where the logic synthesis tool is not constrained by timing considerations, a constraint that further complicates the optimization problem to be performed, the timing associated with synthesized circuit may be unpredictable. The unpredictable timing associated with the synthesized static logic circuit may make the circuit difficult to use with other components of an overall system.

SUMMARY OF THE INVENTION

It is an object of the invention to provide a logic circuit design method which employs simplified logic synthesis. Another object of the invention is to provide a method of logic synthesis which utilizes a dynamic circuit library to simplify the optimization problem which the tool must solve and thereby reduce the processing time required for logic synthesis.

These objects are accomplished by limiting the library available for logic synthesis to a single dynamic circuit block or logic synthesis block. The circuit design method according to the invention includes first defining the logic synthesis block and then performing logic synthesis for a predetermined logical function to be implemented. The synthesis step may be performed by any suitable logic synthesis technique, in particular techniques performed by logic synthesis software tools, and results in an intermediate circuit design. This intermediate circuit design necessarily comprises a series of dynamic circuit blocks, each associated with a single reset signal. The invention then includes eliminating unnecessary devices from this intermediate circuit to produce a final logic circuit, and then sizing the devices in the final circuit to complete the design.

The preferred logic synthesis block comprises the largest practical dynamic AND/OR circuit. The maximum size of this preferred AND/OR circuit is determined generally by the technology in which the circuit is to be implemented. One preferred form of the invention defines a four high and four wide AND/OR block utilizing a single reset signal for the entire block. Regardless of the logic synthesis block defined for the logic synthesis step, the logic synthesis step is preferably performed without constraining the size of the various devices making up the logic synthesis block circuit. Sizing the various devices in the synthesized logic circuit to ensure that the circuit consistently provides the desired logical operation is left as a separate step performed after logic synthesis.

The method according to the invention allows the circuit designer to take advantage of dynamic logic circuitry to improve overall performance of the desired logic circuit. The resulting delayed-reset dynamic logic circuit having a fixed number of reset levels is relatively easy to construct and ensure proper operation as compared to generic dynamic circuits. Perhaps most importantly, limiting the logic synthesis step to a circuit library consisting of a single logic circuit greatly simplifies the logic synthesis step. The remaining steps to finalize the circuit design, that is, the steps of eliminating unnecessary or unused devices in the synthesized intermediate circuit and then sizing the devices in the final circuit, are straightforward tasks which can be done manually or by some automated means.

These and other objects, advantages, and features of the invention will be apparent from the following description of the preferred embodiments, considered along with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 comprises a flowchart showing a circuit design method embodying the principles of the invention.

Figure 2 is an electrical schematic diagram of a prior art four high and four wide unfooted dynamic AND/OR circuit.

Figure 3 is an electrical schematic diagram of a prior art four high and four wide footed dynamic AND/OR circuit.

Figure 4 is a diagrammatic representation of an assembly of logic synthesis blocks resulting from logic synthesis performed according to the invention.

Figure 5 is an electrical schematic diagram of one logic synthesis block included in the assembly shown in Figure 4.

Figure 6 is an electrical schematic diagram showing the final circuit corresponding to the respective logic synthesis block after removing unused devices from the circuit shown in Figure 5.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Figure 1 illustrates the process steps included in a logic circuit design process embodying the principles of the invention. The design process involves a logic synthesis component shown in dashed box 12 and a design component shown in dashed box 14. Steps in logic synthesis component 12 produce an intermediate logic circuit design, described below with reference to Figure 4, which performs a desired logical operation. Steps in design component 14 use this intermediate logic circuit design to provide an efficient final logic circuit for performing the desired logical operation.

The process starts with the designer providing or defining the high level logical operation to be performed by the circuit. According to the invention, logic synthesis

component 12 includes first defining a logic synthesis block as shown at process block 16. The process next includes performing logic synthesis as shown at process block 17 to produce the intermediate circuit capable of performing the predetermined logical operation. According to the invention, this logic synthesis step 17 is constrained to a circuit library including only the logic synthesis block.

Defining the logic synthesis block for use in logic synthesis according to the invention preferably includes defining the largest practical dynamic AND/OR circuit which may be used in the technology in which the circuit is to implemented. This dynamic circuit is controlled by a single reset signal. The largest practical dynamic AND/OR circuit may, for example, comprise a four high and four wide dynamic AND/OR circuit. This four high and four wide AND/OR circuit will be referred to further in this disclosure and shown in the drawings as a "4A4O" circuit.

Figures 2 and 3 show examples of dynamic 4A4O circuits. Each of these circuits is capable of receiving up to four input signals at each of four groups of N-type devices. The first group of four N-type devices is shown in dashed box 21 in Figures 2 and 3, while the remaining groups are shown at dashed boxes 22, 23, and 24. The inputs of the devices included in a single group 21, 22, 23, or 24 are ANDed and then ORed with inputs at the other groups. Each circuit includes a reset device 25 and and output inverter 26 and feedback device 27. Figure 2 shows an unfooted 4A4O circuit while Figure 3 shows a 4A4O circuit including a footer device 28 which also receives the reset signal. 4A4O

circuits are preferred for use as the logic synthesis block according to invention because it is possible to produce a large number of logical combinations from the circuit. The 4A4O logic block also takes advantage of the maximum practical stack height, that is, the maximum number of devices that may be stacked between V_{dd} and ground, which may currently be implemented.

Those skilled in the art will appreciate that the input signals to the devices in groups 21, 22, 23, and 24 are logic level signals comprising either a high logic level signal representing one binary value or a low logic level signal representing the opposite binary value. The actual voltage of the various signals will depend upon the implementation technology. The invention is not limited to any particular high or low level logic signals or any implementation technology. Furthermore, the invention is not limited to using 4A4O circuits as the logic synthesis blocks. These 4A4O logic synthesis blocks simply represent one preferred form of the invention.

The result or output of logic synthesis at step 17 in Figure 1 is an intermediate circuit 29 comprising an assembly of at least one string 30 of logic synthesis blocks 33, 34, 35, and 36. Such an assembly having an indeterminate number of strings 30 is shown in Figure 4. The logic synthesis blocks in the lowermost string 30i in the drawing are labeled as 33i, 34i, 35i, and 36i to indicate that the string is the last of some indeterminate number of strings. Each logic synthesis block 33 to 33i, 34 to 34i, 35 to 35i, or 36 to 36i in each string 30 to 30i comprises the circuit defined as the logic synthesis block for purposes of

performing logic synthesis at step 17. Since the logic synthesis step according to the invention may only select a single type of circuit to combine in some manner to produce the predefined logical operation, the result of logic synthesis at step 17 in Figure 1 will inevitably result in some combination of strings 30 to 30i which include only the defined logic synthesis block. As will be discussed further below however, not all of the functionality of each logic synthesis block will be used to provide the desired logical operation.

The invention will be described further with reference to the preferred logic synthesis block comprising a 4A4O circuit as described above, and in particular a footed 4A4O circuit such as shown in Figure 3. It will be appreciated, however, that the invention is not limited to this particular circuit for use as the logic synthesis block.

In the example shown in Figure 4, each string 30 to 30i of logic synthesis blocks includes four levels of 4A4O blocks. The level one blocks for the various strings comprise blocks 33 to 33i, the level two blocks comprise blocks 34 to 34i, the level three blocks comprise blocks 35 to 35i, and the level four blocks comprise blocks 36 to 36i. Level one block 33 to 33i will typically comprise the footed version of the defined logic synthesis block while the blocks on the following levels, such as blocks 34 to 34i, 35 to 35i, and 36 to 36i in the example, may comprise the unfooted version of the defined logic synthesis block. It will be appreciated that there will be a characteristic delay for the reset signal provided to each successive logic synthesis block level. The delay between the reset signal

applied to the level one blocks 33 to 33i and the level two blocks 34 to 34i is represented by delay element 40. The delays associated with the level three blocks 35 to 35i and level four blocks 36 to 36i, are represented by delay elements 41 and 42, respectively. These delays are required to allow the desired data to propagate through the various logic synthesis blocks included in the assembly and provide the desired output for application to the next succeeding block.

The logic synthesis step shown at process block 17 may be performed in any suitable fashion. In particular, the logic synthesis step may be performed by a logic synthesis software tool executed by a suitable processing device. The Synopsys Design Compiler™ logic synthesis tool by Synopsys, Inc. of Mountain View, California is an example of a logic synthesis software tool which may be employed to perform the logic synthesis step according to the invention.

Although the library available for logic synthesis is constrained to the single logic synthesis block, the preferred form of the invention includes leaving the size of devices included in the logic synthesis block substantially unconstrained. That is, rather than including in the logic synthesis library a number of the logic synthesis blocks each with a different combination of device sizes, the logic synthesis step is performed with a generic logic synthesis block. This preferred generic logic synthesis block is assumed to be capable of meeting drive requirements and providing the desired logical operations based

upon the general or generic operational characteristics of the various devices included in the circuit.

Referring back to Figure 1, the design component processes shown at dashed box 14 use the intermediate circuit produced through logic synthesis step 17 to produce a final circuit. It will be unlikely that the intermediate circuit 29 shown in Figure 4 will actually require all of the functionality available in each 4A4O circuit, logic synthesis block 33 to 33i, 34 to 34i, 35 to 35i, and 36 to 36i included in the intermediate circuit. At least some of the devices included in the various logic synthesis blocks 33 to 33i, 34 to 34i, 35 to 35i, and 36 to 36i will be unused in effecting the final logical operation performed by the intermediate circuit. Thus, the design component processes shown in dashed box 14 preferably include determining if there are unused devices in the intermediate circuit 29. This step is shown at decision box 45 in Figure 1. If there are unused devices in the intermediate circuit 29 made up of logic synthesis blocks, the process includes eliminating those unused or unnecessary devices as shown at process block 46 in Figure 1. The steps of detecting unused devices in intermediate circuit 29 and eliminating those unused devices may be described with reference to a simple example shown in Figures 5 and 6.

Figure 5 shows a single level, for example the level one 4A4O block 33 from Figure 4. It is assumed for the purposes of this example that the synthesized intermediate circuit 29 which includes first level block 33 simply ANDs two input signals to that particular block and that the result of the AND operation is output from the first level block

to second level block 34 shown in Figure 4. In this case, the logic synthesis step selects the first group of devices 21 in the block, and particularly devices 51 and 52 for receiving the input signals d_1 and d_2. The logic synthesis step performed at block 17 also must specify the inputs to devices 53 and 54 in that group of devices to each receive a high-level logic signal so as not to interfere with the AND operation provided through devices 51 and 52. Devices in the other groups, 22, 23, and 24 would each receive a common level signal, preferably a low level signal since they are not relevant in providing the desired AND function in the first level logic block 33. Since devices 53 and 54 as well as all of the devices in groups 22, 23 and 24 do not affect the logical operation provided by the intermediate circuit in which they are included, all of these devices may be removed from the circuit without affecting the operation of the circuit. That is, devices 53 and 54 and all devices in groups 22, 23, and 24 may be removed from the intermediate circuit without affecting the desired AND function provided through devices 51 and 52.

The result of the logic synthesis step shown at process block 17 in Figure 1 will indicate any devices in the various logic synthesis blocks 33 to 33i, 34 to 34i, 35 to 35i, and 36 to 36i included in intermediate circuit 29 which do not change state in the operation of the circuit. Removing these unused or unnecessary devices from intermediate circuit 29 is thus preferably performed by detecting any devices in the synthesized intermediate circuit which do not change state in the operation of the circuit, and then eliminating those detected devices from the intermediate circuit. These steps may be performed manually or

in some automated fashion through a suitable software tool operating on the result or output of logic synthesis step 17 in Figure 1.

Regardless of how unused devices are detected and removed from the intermediate circuit design, the resulting circuit may be referred to as a final circuit or final circuit design. Figure 6 shows the portion of the final circuit remaining from block 33 in our example described above with reference to Figure 5. This first level portion of the final circuit includes only devices 51 and 52, along with reset device 25, footer device 28, output inverter 26, and feedback/pull-up device 27.

Once there are no more unused or unnecessary devices to remove from the intermediate circuit, the present circuit design process includes the step of analyzing the circuit to determine the size of devices remaining in the final circuit necessary to provide the desired functionality. This sizing step is shown at process block 48 in Figure 1, and involves determining the drive strength required of the various devices. For example, devices 51 and 52 in the AND circuit shown in Figure 6 may be sized differently than devices included in a full group such as group 22 in Figure 5, should the full group of devices remain in the final circuit. Determining the size of the various devices in a logic circuit in order to provide the desired logic function for a given supply voltage and other operating conditions is well known in the field of circuit design and will not be discussed here in further detail.

The logic circuit design method according to the invention drastically simplifies the optimization problem required to perform logic synthesis. Thus, the processing time required for synthesizing complex logic circuitry is greatly reduced by the invention. Reset signal generation and timing is also greatly simplified using a fixed number of levels of delayed-reset dynamic circuit elements. Furthermore, leaving the device size in the logic synthesis block unconstrained as in the preferred form of the invention allows the designer to set device size to keep a fairly constant delay for each block at each level in the final circuit. This in turn allows unfooted blocks to be used in subsequent levels of the final circuit without running into reset timing problems.

The above described preferred embodiments are intended to illustrate the principles of the invention, but not to limit the scope of the invention. Various other embodiments and modifications to these preferred embodiments may be made by those skilled in the art without departing from the scope of the following claims. For example, although the synthesized intermediate circuit 29 is described as having four dynamic block levels for purposes of example, the invention is not limited to a four level dynamic circuit. Also, the invention is not limited to any particular logic synthesis technique as long as the library of circuits available in the technique is limited to a single logic circuit or logic synthesis block.